WHAT IS CLAIMED IS:

1. A pipeline accelerator, comprising:

a memory; and

a hardwired-pipeline circuit coupled to the memory and operable to,

receive data,

load the data into the memory,

retrieve the data from the memory,

process the retrieved data, and

provide the processed data to an external source.

2. The pipeline accelerator of claim 1 wherein:

the memory is disposed on a first integrated circuit; and

the pipeline circuit is disposed on a second integrated circuit.

3. The pipeline accelerator of claim 1 wherein the pipeline circuit is disposed on a field-programmable gate array.

4. The pipeline accelerator of claim 1 wherein the pipeline circuit is operable to provide the processed data to the external source by:

loading the processed data into the memory;

retrieving the processed data from the memory; and

providing the retrieved processed data to the external source.

5. The pipeline accelerator of claim 1 wherein:

the external source comprises a processor; and

the pipeline circuit is operable to receive the data from the processor.

6. A computing machine, comprising:

a processor; and

a pipeline accelerator coupled to the processor and comprising,

a memory, and

a hardwired-pipeline circuit coupled to the memory and operable to,

receive data from the processor,

load the data into the memory,

retrieve the data from the memory,
process the retrieved data, and
provide the processed data to the processor.

7. A pipeline accelerator, comprising:
a memory; and
a hardwired-pipeline circuit coupled to the memory and operable to,
receive data,
process the received data,
load the processed data into the memory,
retrieve the processed data from the memory, and
provide the retrieved processed data to an external source.

8. A computing machine, comprising:
a processor; and
a pipeline accelerator coupled to the processor and comprising,
a memory, and
a hardwired-pipeline circuit coupled to the memory and operable to,
receive data from the processor,
process the received data,
load the processed data into the memory,
retrieve the processed data from the memory, and
provide the retrieved processed data to the processor.

9. A pipeline accelerator, comprising:
first and second memories; and
a hardwired-pipeline circuit coupled to the first and second memories and comprising,
an input-data handler operable to receive raw data from an external source and to load the raw data into the first memory,
a hardwired pipeline operable to process the raw data,

a pipeline interface operable to retrieve the raw data from the first memory, provide the retrieved raw data to the hardwired pipeline, and load processed data from the hardwired pipeline into the second memory, and

an output-data handler operable to retrieve the processed data from the second memory and to provide the processed data to the external source.

10. The pipeline accelerator of claim 9 wherein:

the first and second memories each include respective first and second ports;

the input-data handler is operable to load the raw data via the first port of the first memory,

the pipeline interface is operable to retrieve the raw data via the second port of the first memory and to load the processed data via the first port of the second memory, and

the output-data handler is operable to retrieve the processed data via the second port of the second memory.

11. The pipeline accelerator of claim 9, further comprising:

a third memory coupled to the hardwired-pipeline circuit;

wherein the hardwired pipeline is operable to generate intermediate data while processing the raw data; and

wherein the pipeline interface is operable to load the intermediate data into the third memory and to retrieve the intermediate data from the third memory.

12. The pipeline accelerator of claim 9 wherein:

the first and second memories are respectively disposed on first and second integrated circuits; and

the pipeline circuit is disposed on a field-programmable gate array.

13. The pipeline accelerator of claim 9, further comprising:

an input-data queue coupled to the input-data handler and the pipeline interface;

wherein the input-data handler is operable to load into the input-data queue a pointer to a location of the raw data within the first memory; and

wherein the pipeline interface is operable to retrieve the raw data from the location using the pointer.

14. The pipeline accelerator of claim 9, further comprising:

an output-data queue coupled to the output-data handler and the pipeline interface;

wherein the pipeline interface is operable to load into the output-data queue a pointer to a location of the processed data within the second memory; and

wherein the output-data handler is operable to retrieve the processed data from the location using the pointer.

15. The pipeline accelerator of claim 9, further comprising:

wherein each of the input-data handler, hardwired pipeline, pipeline interface, and output-data handler has a respective operating configuration; and

a configuration manager coupled to and operable to set the operating configurations of the input-data handler, hardwired pipeline, pipeline interface, and output-data handler.

16. The pipeline accelerator of claim 9, further comprising:

wherein each of the input-data handler, hardwired pipeline, pipeline interface, and output-data handler has a respective operating status; and

an exception manager coupled to and operable to identify an exception in the input-data handler, hardwired pipeline, pipeline interface, or output-data handler in response to the operating statuses.

17. A pipeline accelerator, comprising:

a hardwired pipeline operable to process data; and

an input-data handler coupled to the hardwired pipeline and operable to,

receive the data,

determine whether the data is directed to the hardwired pipeline, and

provide the data to the hardwired pipeline if the data is directed to the hardwired pipeline.

18. The pipeline accelerator of claim 17 wherein the input-data handler is further operable to:

receive the data by,

receiving a message that includes a header and the data, and

extracting the data from the message; and

determine whether the data is directed to the hardwired pipeline by analyzing the header.

19. The pipeline accelerator of claim 17 wherein the hardwired pipeline and the input-data handler are disposed on a single field-programmable gate array.

20. The pipeline accelerator of claim 17 wherein the hardwired pipeline and the input-data handler are disposed on respective field-programmable gate arrays.

21. A computing machine, comprising:

a processor; and

a pipeline accelerator coupled to the processor and comprising,

a hardwired pipeline operable to process data, and

an input-data handler coupled to the hardwired pipeline and operable to,

receive the data from the processor,

determine whether the data is directed to the hardwired pipeline, and

provide the data to the hardwired pipeline if the data is directed to the hardwired pipeline.

22. A pipeline accelerator, comprising:

a hardwired pipeline operable to generate data; and

an output-data handler coupled to the hardwired pipeline and operable to,

receive the data,

determine a destination of the data, and

provide the data to the destination.

23. The pipeline accelerator of claim 22 wherein the output-data handler is further operable to:

determine the destination of the data by,

identifying a type of the data, and

determining the destination based on the type of the data; and

provide the data to the destination by,

generating a message that identifies the destination and that includes the data, and

providing the message to the destination.

24. A computing machine, comprising:

a processor operable to execute threads of an application; and

a pipeline accelerator coupled to the processor and comprising:

a hardwired pipeline operable to generate data, and

an output-data handler coupled to the hardwired pipeline and operable to,

receive the data,

identify a thread of the application that subscribes to the data, and

provide the data to the subscribing thread.

25. A pipeline accelerator, comprising:

a hardwired pipeline operable to process data values; and

a sequence manager coupled to and operable to control the operation of the hardwired pipeline.

26. The pipeline accelerator of claim 25 wherein the sequence manager is operable to control an order in which the hardwired pipeline receives the data values.

27. The pipeline accelerator of claim 25 wherein the sequence manager is further operable to:

receive an event; and

control the hardwired pipeline in response to the event.

28. The pipeline accelerator of claim 25 wherein the sequence manager is further operable to:

receive a synchronization signal; and

control the operation of the hardwired pipeline in response to the synchronization signal.

29. The pipeline accelerator of claim 25 wherein the sequence manager is further operable to:

sense an occurrence relative to the hardwired pipeline; and

generate an event in response to the occurrence.

30. A computing machine, comprising:

a processor operable to generate data and an event; and

a pipeline accelerator coupled to the processor and comprising,

a hardwired pipeline operable to receive the data from the processor and process the received data; and

a sequence manager coupled to the hardwired pipeline and operable to receive the event from the processor and to control the operation of the hardwired pipeline in response to the event.

31. A pipeline accelerator, comprising:

a hardwired-pipeline circuit having an operating configuration and operable to process data; and

a configuration manager coupled to the hardwired-pipeline circuit and operable to set the operating configuration.

32. The pipeline accelerator of claim 31 wherein:

the hardwired-pipeline circuit includes a configuration register; and

the configuration manager is operable to set the operating configuration by loading a configuration value into the configuration register.

33. The pipeline accelerator of claim 32 wherein the configuration manager is operable to receive the configuration value from an external source.

34. A computing machine, comprising:

a processor operable to generate data and a configuration value; and

pipeline accelerator coupled to the processor and comprising,

a hardwired-pipeline circuit having an operating configuration and operable to process the data, and

a configuration manager coupled to the hardwired-pipeline circuit and operable to set the operating configuration in response to the configuration value.

35. A pipeline accelerator, comprising:

a hardwired-pipeline circuit having an operating status and operable to process data; and

an exception manager coupled to the hardwired-pipeline circuit and operable to identify an exception in the operation status of the hardwired-pipeline circuit in response to the operating status.

36. The pipeline accelerator of claim 35 wherein:

the hardwired-pipeline circuit is operable to generate a status value that represents the operating status; and

the exception manager is operable to identify the exception in response to the status value.

37. The pipeline accelerator of claim 36 wherein:

the hardwired-pipeline circuit includes a status register that is operable to store the status value; and

the exception manager receives the status value from the status register.

38. The pipeline accelerator of claim 35 wherein the exception manager is operable to identify an exceptionin the operating status of the hardwired-pipeline circuit to an external source.

39. A computing machine, comprising:

a processor operable to generate data; and

a pipeline accelerator, comprising,

a hardwired-pipeline circuit having an operating status and operable to process data and to generate a status value that represents the operating status, and

an exception manager coupled to the hardwired-pipeline circuit and operable to identify an exception in the operating status of the hardwired-pipeline circuit in response to the status value and to notify the processor of the exception.

40. A computing machine, comprising:

a pipeline accelerator, comprising,

a hardwired-pipeline circuit having an operating status and operable to process data, and

an exception manager coupled to the hardwired-pipeline circuit and operable to generate a status value that represents the operating status; and

a processor coupled to the pipeline accelerator and operable to generate the data, to receive the status value, and to determine whether the hardwired-pipeline circuit is malfunctioning by analyzing the status value.

41. A method, comprising:

loading data into a memory;

retrieving the data from the memory;

processing the retrieved data with a hardwired-pipeline circuit; and

providing the processed data to an external source.

42. The method of claim 41 wherein providing the processed data comprises:

loading the processed data into the memory;

retrieving the processed data from the memory; and

providing the retrieved processed data to the external source.

43. A method, comprising:

processing data with a hardwired-pipeline circuit;

loading the processed data into a memory;

retrieving the processed data from the memory; and

providing the retrieved processed data to an external source.

44. A method, comprising:

loading raw data from an external source into a first memory;

retrieving the raw data from the first memory;

processing the retrieved data with a hardwired pipeline;

loading the processed data from the hardwired pipeline into a second memory; and

providing the processed data from the second memory to the external source.

45. The method of claim 44 wherein:

loading the raw data comprises loading the raw data via a first port of the first memory;

retrieving the raw data comprises retrieving the raw data via a second port of the first memory;

loading the processed data comprises loading the processed data via a first port of the second memory; and

providing the processed data comprises retrieving the processed data via a second port of the second memory.

46. The method of claim 44, further comprising:

generating intermediate data with the hardwired pipeline in response to processing the raw data;

loading the intermediate data into a third memory; and

providing the intermediate data from the third memory back to the hardwired pipeline.

47. The method of claim 44, further comprising:

loading into an input-message queue a pointer to a location of the raw data within the first memory; and

wherein retrieving the raw data comprises retrieving the raw data from the location using the pointer.

48. The method of claim 44, further comprising:

loading into an output-message queue a pointer to a location of the processed data within the second memory; and

wherein retrieving the processed data comprises retrieving the processed data from the location using the pointer.

49. The method of claim 44, further comprising setting parameters for loading and retrieving the raw data, processing the retrieved data, and loading and providing the processed data.

50. The method of claim 44, further comprising determining whether an error occurs during the loading and retrieving of the raw data, the processing of the retrieved data, and the loading and providing of the processed data.

51. A method, comprising:

receiving data;

determining whether the data is directed to a hardwired pipeline; and

providing the data to the hardwired pipeline if the data is directed to the hardwired pipeline.

52. The method of claim 51 wherein:

receiving the data comprises,

receiving a message that includes a header and the data, and

extracting the data from the message; and

determining whether the data is directed to the hardwired pipeline comprises analyzing the header.

53. A method, comprising:

generating data with a hardwired pipeline;

determining a destination of the data; and

providing the data to the destination.

54. The method of claim 53 wherein:

determining the destination of the data comprises,

identifying a type of the data, and

determining the destination based on the type of the data; and

providing the data to the destination comprises,

generating a message that identifies the destination and that includes the data, and

providing the message to the destination.

55. A method, comprising:

processing data values with a hardwired pipeline; and

sequencing the operation of the hardwired pipeline.

56. The method of claim 55 wherein sequencing the operation comprises sequencing an order in which the hardwired pipeline processes the data values.

57. The method of claim 55 wherein sequencing the operating comprises synchronizing the operation of the hardwired pipeline to a synchronization signal.

58. The method of claim 55, further comprising:

sensing a predefined occurrence during operation of the hardwired pipeline; and

generating an event in response to the occurrence.

59. A method, comprising:

loading a configuration value into a register; and

setting an operating configuration of a hardwired pipeline with the configuration value.

60. A method, comprising:

processing data with a hardwired pipeline; and

identifying an error in the processed data by analyzing an operating status of the hardwired pipeline.

61. A method for designing a hardwired-pipeline circuit, comprising:

retrieving from a library a first data representation of a communication interface;

generating a second data representation of a hardwired pipeline that is to be coupled to the communication interface; and

combining the first and second data representations to generate hard-configuration data for the hardwired-pipeline circuit.

62. The method of claim 61, further comprising modifying the first data representation by selecting values for predetermined parameters of the services layer before combining the first and second data representations.

63. The method of claim 61 wherein the communication interface is operable to allow the hardwired-pipeline circuit to communicate with another circuit.

64. The method of claim 61 wherein combining the first and second data representations comprises compiling the first and second data representations into the hard-configuration data.

65. The method of claim 61 wherein the hard-configuration data comprises firmware.